Institut de la Francophonie pour Innovation

Conception et architectures des réseaux

Travail Pratique

Linux en réseau

A l'issue de ce TP vous devez:

- connaître et savoir utiliser les commandes de base de Unix pour configurer et tester les connexions réseau;
- savoir configurer en réseau un poste de travail sous Linux sans faire recours aux outils graphiques;
- savoir analyser les protocoles de communication à l'aide des programmes pour la capture des trames;
- être capable de concevoir des petites applications en réseau IP en utilisant la programmation par sockets.

Outils réseau sous Linux

Linux contient de nombreux utilitaires permettant de faciliter la configuration et l'administration des interfaces réseau ainsi que de tester certains aspects du fonctionnement du réseau.

Pour plus d'information sur les utilitaires, leur mode d'utilisation et les options correspondantes a utiliser les pages de manuel:
`man <nom_utilitaire>` ou
`info <nom_utilitaire>`.

Outils de configuration

Interface réseau: l'interface réseau est représentée physiquement par votre carte réseau mais ce terme est aussi utilisé pour désigner un nom logiciel auquel assigner une adresse IP (eth0 par exemple). Une adresse IP est toujours assignée à une interface réseau, jamais à un ordinateur. La commande *ifconfig* sert à afficher la configuration des différentes interfaces réseau actives et, en même temps, de les configurer (a faire attention au fait que la configuration sera perdue lors du redémarrage de la machine).

`ifconfig` : utilitaire standard de UNIX permettant d'obtenir des informations sur la configuration de l'interface réseau (carte Ethernet par exemple) :

```
$ ifconfig -a
```

Servez-vous de **`man ifconfig`** pour connaître les options de cette commande.

Les interfaces réseau d'Ubuntu sont marquées d'habitude avec *eth0, eth1*, etc… pour le réseau filaire et avec *wlan0, wlan1*, etc. pour le réseau wifi.

`route` : modifier le table de routage

Fichiers de configuration :

/etc/hosts: ce fichier spécifie comment résoudre les noms des machines du réseau local (inutile de mettre en œuvre un serveur DNS pour un petit réseau local). La syntaxe des lignes de ce fichier est:

```
Adresse IP      Nom de l'hôte        Alias
```

Exemple :

```
 127.0.0.1      localhost
 192.168.0.1    sirius.ifi.edu.vn    sirius
```

/etc/resolv.conf: ce fichier spécifie où résoudre ce qui ne se trouve pas dans /etc/hosts. C'est dans ce fichier que vous devez spécifier les adresses IP des serveurs DNS utilisés pour accéder à Internet (ainsi que des certaines options pour la résolution de nom) en suivant la syntaxe (minimale)

suivante:

```
nameserver 192.168.102.1
```

Pour plus d'information sur le format et les options disponibles dans ce fichier, faites un **`man resolv.conf`**

/etc/hostname: ce fichier configure le nom de la machine locale. Au démarrage du système, ce fichier est lu et son contenu est envoyé à la commande hostname. Vous pouvez utiliser la commande hostname pour changer le nom de votre machine.

/etc/network/interfaces: (pour Debian et Ubuntu) configuration des interfaces réseau. Pour les autres distributions Linux il faut regarder la documentation**.**

Outils pour diagnostiquer le réseau

`ping` : l'outil le plus simple et le plus pratique des outils réseaux. Ping permet de vérifier si un nom d'hôte distant ou une adresse IP est accessible.

`traceroute (tracepath, tcptraceroute)`: utilitaire pour diagnostiquer des problèmes réseaux, en particulier si la commande ping ne réussit pas à atteindre le serveur distant. Il existe des outils avec des fonctionnalités de traceroute et avec des interfaces utilisateur plus évoluées (ex: **`mtr`**) ou même graphiques (**`xtraceroute`**, par exemple) permettant de visualiser le chemin parcouru par les données entre le client et le serveur.

`netstat` : utilitaire très complet pour afficher la configuration réseau ainsi que les statistiques sous les systèmes UNIX.

`dig, host` et **`nslookup`** : outils pour interroger les serveurs de nom (résolution de nom).

Outils pour utiliser les services réseau

`telnet` – client pour accéder à distance à une machine UNIX avec une connexion non-sécurisée: le mot de passe et les données sont transférées en clair.

`ftp` - client pour télécharger des fichiers avec une connexion non-sécurisée.

`telnet` ***et*** **`ftp`** sont des applications qui utilisent une communication en clair (c'est à dire sans aucune protection de données), il existe des outils sécurisés équivalents qui assurent les mêmes fonctionnalités (vous êtes fortement encouragés de les utiliser pour garantir la sécurité de vos communications) et éviter en même temps d'utiliser des outils non-sécurisés.

`ssh` - client pour accéder à distance à une machine avec une connexion sécurisée: tout échange de données est cryptée.

`scp` et **`sftp`** – utilitaires pour échanger des fichiers avec une machine distante à travers une connexion sécurisée.

Travail à faire:

Découvrez les utilitaires, leur mode d'utilisation et les options correspondantes, analysez les fichiers de configuration de votre machine. (Attention: il ne s'agit pas de citer les outils graphiques dont dispose Gnome ou KDE mais les commandes Linux et les fichiers de configuration!). Certaines applications (paquets) ne sont pas installées par défaut, vous pouvez le faire en ligne de commande (sous Debian et Ubuntu):

```
aptitude install <nom paquet>
```

La plupart des distributions actuelles de Linux proposent des interfaces graphiques pour l'installation de paquets, je vous laisse les découvrir par vous mêmes.

Testez chacune de ces commandes les unes après les autres et donnez :

- la liste des interfaces sur votre machine ;
- l'adresse IP de votre machine ;
- l'adresse MAC de votre carte réseau ;
- l'adresse et le masque de votre réseau ;
- la table de routage de votre machine ;
- le nom de la machine d'adresse IP 112.137.140.41, son domaine et le serveur de nom de son domaine ;
- donnez la liste des routeurs par lesquels passent les datagrammes entre vous et la machine 112.137.140.41
- trouver le(s) serveur(s) de nom pour les domaines *fpt.com.vn* et *ifi.edu.vn* (le domaine de l'IFI)

A découvrir par vous mêmes : comment configurer une interface wifi sous Linux sans interface graphique.

Outils pour la capture des trames

Les outils pour la capture et l'analyse des trames sont indispensables pour tout développement de protocole. La capture de trames permet d'analyser le bon fonctionnement des différents protocoles ainsi que les différents temps d'exécution de ces protocoles. Vous trouverez en annexe un brève rappel sur la structure des unités de données dans les réseaux TCP/IP. Pour plus d'information a regarder les RFC correspondants.

Linux dispose de l'utilitaire *tcpdump* qui permet de capturer des trames en mode textuel et *Wireshark* (également connu jusqu'en 2006 comme *Ethereal*) - un logiciel libre, assez puissant (disponible sur un grand nombre de plates-formes matérielles et de systèmes d'exploitation) pour capturer et analyser les trames en mode graphique. Vous pouvez télécharger la dernière version de ce logiciel sur www.wireshark.org, il est également disponible dans les dépôts des distributions Linux sous forme précompilée (**`aptitude install wireshark wireshark-doc`** pour l'installer). Vous trouverez une brève description du mode d'utilisation de ce logiciel en suivant les liens sur la page de cours.

Travail à faire:

Analyse du protocole de résolution d'adresse ARP

Démarrez un capture en ayant préalablement paramétré vos filtres afin de ne visualiser que les trames qui vous intéressent. Consultez le cache ARP de votre machine (commande `arp -a`). Pinguez une adresse IP de votre réseau local qui ne soit pas dans le cache ARP de votre machine. En analysant les trames capturées, déterminez le fonctionnement du protocole ARP. Pour ce faire, consultez en détail la couche liaison (Ethernet) des trames et comparez en détail le contenu de la trame au niveau ARP/RARP entre le *Reply* et le *Request*.

Analyse des routes suivies par les paquets (l'outil `mtr`)

Le but de cet exercice est de découvrir le fonctionnement de l'outil **`mtr`** qui permet de suivre les routes empruntées par les paquets qui circulent entre deux hôtes sur Internet. Faites **`aptitude install mtr-tiny`** pour l'installer s'il n'est pas présent sur votre poste. Lancez le logiciel pour tracer la route vers le site www.vnpt.com.vn (ou un autre si vous le désirez). Il existent d'autres logiciels capables de le faire : `traceroute`. `tcptraceroute`, `tracepath`, etc. (vous pouvez les essayer aussi). Recherchez au niveau IP, les champs qui varient entre les envois successifs de paquets (excepté le champ identification et checksum qui ne sont d'aucun intérêt). Que remarquez-vous sur la succession des adresses IP affichées par rapport à l'ensemble des interfaces réseaux réellement traversées pour joindre la destination ? (*Aide:* en étudiant la topologie du réseau et plus particulièrement le plan d'adressage, déterminez la succession des interfaces traversées par vos paquets). Expliquez le fonctionnement de ce logiciel en vous basant *exclusivement* sur la capture de trames.

Analyse du protocole TCP

On vous propose de faire une analyse détaillée du protocole TCP par capture de trames. Vous allez télécharger sur votre machine un fichier avec un protocole qui utilise une connexion TCP et capturer en même temps tous les paquets qui concernent cette communication. On utilisera le protocole http pour télécharger sur votre machine un fichier à partir du serveur intranet.dorsale.ifi:

- Lancer la capture avec tcpdump dans un fichier (pensez a filtrer les paquets capturés pour avoir que les informations qui concernent la connexion correspondante). Donnez la commande que vous avez utilisée.
- Lancer la commande (installez prealablement wget : **`aptitude install wget`**) :

```
wget http://fad.ifi.edu.vn/ififad/file.php/28/documents/WS_user-guide-a4.pdf
```

- Arrêtez la capture et analysez les résultats:
  - Donnez le listing de la capture dans le rapport et identifiez les zonez qui correspondent aux phases de connexion, transfert de données, déconnexion;
  - Identifiez les numéros initiaux de séquence (Initial Sequence Number - ISN) utilisés dans les deux sens;
  - Représentez sur un schéma cette échange en montrant l'évolution des numéros de séquence, de la fenêtre, des acquittements, des fanions, etc.

Analyse du protocole telnet et la capture des informations qui circulent en clair sur le réseau

Cette exercice vise a vous montrer le danger d'utiliser des protocoles non-securisés qui utilisent des échanges en clair. Installez (**`apt-get install telnetd`**) et lancez le serveur telnet sur une des machines de votre équipe. Ajoutez un nouveau utilisateur (avec le nom *test,* par exemple). Lancez la capture de trames, faites un telnet sur cette machine, connectez-vous avec le nom *test* et son mot de passe. Évitez les fautes de frappes qui génèrent du trafic inutile. Arrêtez et affichez la capture. Étant donné le nombre important des trames capturées, lors de l'analyse des trames prenez seulement en compte les trames dont les segments TCP encapsulent des données.

- Indiquez les ports TCP source et destination utilisés par telnet (convertir le format hexadécimal en base 10);
- Retrouvez les mots de passe que vous avez utilisé avec telnet? Où cherchez-vous ces informations?

Question: Lorsque vous êtes sur un réseau Ethernet, capturez-vous tous les trames qui traversent le réseau? Expliquez. Idem pour un réseau wifi.

Le contenu du rapport (un rapport par équipe)

- Les informations sur les interfaces de votre machine demandées au début de cette partie;
- Explications sur la configuration des interfaces wifi sous Linux;
- Une brève analyse des captures des trames réalisées (*ARP, telnet* etc.) avec les parties les plus concluantes, évidemment). On vous demande une vérification pratique des échanges à l'aide des trames capturées même si vous connaissez en théorie le fonctionnement des logiciels ;
- Explications sur le fonctionnement de l'outil *mtr ;*
- Une analyse détaillée du protocole TCP (capture; analyse, diagramme, etc.)
- Vos conclusions.

Les rapports doivent être déposés dans le dossier correspondant sur la page de module.

La date limite de remise des rapports est sur la page du cours.

Annexe

Structure des unités de données

Structure de la trame Ethernet

```
+--6-octets---+--6-octets---+-2o-+- - - - - - - - - -+
| adresse     | adresse     |type|     données       |
| destination | source      |    |                   |
+-------------+-------------+----+- - - - - - - - - -+
```

Trame présentée sans préambule ni CRC.

Quelques types : 0x0200 = XEROX PUP 0x0800 = IPv4
0x0806 = ARP 0x8035 = RARP
0X814C = SNMP 0x86DD= IPv6
<1500 - 802.3 ...

La liste complète des types des trames Ethernet:
http://www.iana.org/assignments/ethernet-numbers

Structure du paquet IPv4

```
<-----------------32bits---------------------->
<-4b->      <--8bits--><--------16bits-------->
+-----+-----+----------+----------------------+
| Ver | IHL |   TOS    |    Longueur totale   |
+-----+-----+----------+--+-------------------+
| Identificateur        |Fl|      FO           |
+-----------+-----------+--+-------------------+
| TTL       | Protocole | Somme de ctrl (entete)|
+-----------+-----------+-----------------------+
| Adresse Source                               |
+----------------------------------------------+
| Adresse Destination                          |
+----------------------------------------------+
... Options
+----------------------------------------------+
... Données
+----------------------------------------------+
```

Ver = Version d'IP,
IHL = Longueur de l'entête IP (en mots de 32 bits),
TOS = Type de service,
Longueur totale du paquet IP (en octets).
Fl (3 bits) = Indicateurs pour la fragmentation
1er = réservé,
2ème = ne pas fragmenter et
3ème = fragment suivant existe),
FO (13 bits) = décalage du fragment (valeur a multiplier par 8 octets),
TTL = durée de vie restante.

Quelques protocoles transportés :

1 = ICMP	8 = EGP
2 = IGMP	11 = DoP
4 = IP (encapsulation)	17 = UDP
5 = Stream	36 = XTP
6 = TCP	46 = RSVP

Structure du datagramme ICMP

```
<-----------------32bits---------------------->
+-----------+-----------+----------------------+
| Type      | Code      | Somme de contrôle (dtg|
+-----------+-----------+----------------------+
|     Variable (généralement non utilisé)      |
+----------------------------------------------+
...      Datagramme original + 8 octets
+----------------------------------------------+
```

Quelques types ICMP : 0 = Demande d'écho,
8 = Réponse d'écho,
11 = Durée de vie écoulée,
12 = Erreur de paramètre...

Structure ARP

```
+16b-+-16b-+8b+8b+16b+--lgHW--+lgP-+--lgHW--+lgP-+
|type|type |lg|lg|Op |Emetteur|Emt.|Recept. |Rcpt|
|HW  |Proto|HW|P |   |adr. HW |adrP|adr. HW |adrP|
+----+-----+--+--+---+--------+----+--------+----+
```

Quelques types : 0x0001 = Ethernet 0x0800 = DoD Internet
Opérations : 0x0001 = Requête 0x0002 = Réponse

Structure du datagramme UDP

```
<-----------------32bits---------------------->
+-----------------------+-----------------------+
| Port Source           | Port Destination      |
+-----------------------+-----------------------+
| Longueur UDP          |Somme de ctrl (message)|
+-----------------------+-----------------------+
... Données
+----------------------------------------------+
```

Structure du segment TCP

```
<-----------------32bits---------------------->
<-4b->         <-6bits-><--------16bits-------->
+-----------------------+-----------------------+
| Port Source           | Port Destination      |
+-----------------------+-----------------------+
| Numéro de Séquence                            |
+-----------------------------------------------+
| Numéro d'Acquittement                         |
+-----+-------+---------+-----------------------+
| THL |       | Flag    | Taille Fenetre        |
+-----+-------+---------+-----------------------+
| Somme de ctrl (message)    Pointeur d'urgence |
+-----------------------+-----------------------+
... Options
+----------------------------------------------+
... Données
+----------------------------------------------+
```

THL = longueur de l'entête TCP sur 4 bits (en mots de 32 bits)
Flags (6 bits) = indicateurs (du bit de poids le plus faible au plus fort):

* 0 = Fin	(FIN)
* 1 = Synchronisation	(SYN)
* 2 = Réinitialisation	(RST)
* 3 = Données immédiates	(PSH)
* 4 = Acquittement	(ACK)
* 5 = Données urgentes	(URG)

Options = suites d'option codées sur
* 1 octet à 00 = Fin des options
* 1 octet à 01 = NOP (pas d'opération)
* plusieurs octets de type TLV (Type, Longueur, Valeur)
T = un octet de type:
2 Négociation de la taille max. du segment
3 Adaptation de la taille de la fenêtre
4 Autorisation des acquittements sélectifs
8 Estampilles temporelles
L = un octet pour la taille totale de l'option
V = valeur de l'option (sur L-2 octets)

Services associés aux ports

ftp-data	20/tcp	ssh	22/udp
ftp	21/tcp	domain	53/udp
ssh	22/tcp	tftp	69/udp
telnet	23/tcp	www	80/ucp
smtp	25/tcp	pop-3	110/udp
domain	53/tcp	snmp	161/udp
www	80/tcp	snmp-trap	162/udp
imap	143/tcp	imap	143/udp
pop-3	110/tcp	pop-3	110/udp
ntp	123/tcp	ntp	123/udp

La liste complète des numéros des ports:
http://www.iana.org/assignments/port-numbers